(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
20 December 2001 (20.12.2001)

PCT

(10) International Publication Number
WO 01/96990 A2

(51) International Patent Classification⁷: G06F 1/00

(21) International Application Number: PCT/EP01/06816

(22) International Filing Date: 15 June 2001 (15.06.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/594,456 15 June 2000 (15.06.2000) US

(71) Applicant: RAINBOW TECHNOLOGIES, B.V. [NL/NL]; Oliphanteweg 10, NL-1397 Le Rotterdam (NL).

(72) Inventors: **ABBOTT, Shawn, D.**; 305 Pinnacle Ridge Place, RR12, Calgary, Alberta T3E 6W3 (CA). **ANDERSON, Allan, D.**; 11158 Bertha Place, Cerritos, CA 90703 (US). **GODDING, Patrick, N.**; 22665 Shady Grove Circle, Lake Forest, CA 92630 (US). **PUNT, Maarten, G.**; 24942 Paseo Arboleda, Lake Forest, CA 92630 (US). **SOTOODEH, Mehdi**; 17 Paloma Drive, Mission Viejo, CA 92692 (US).

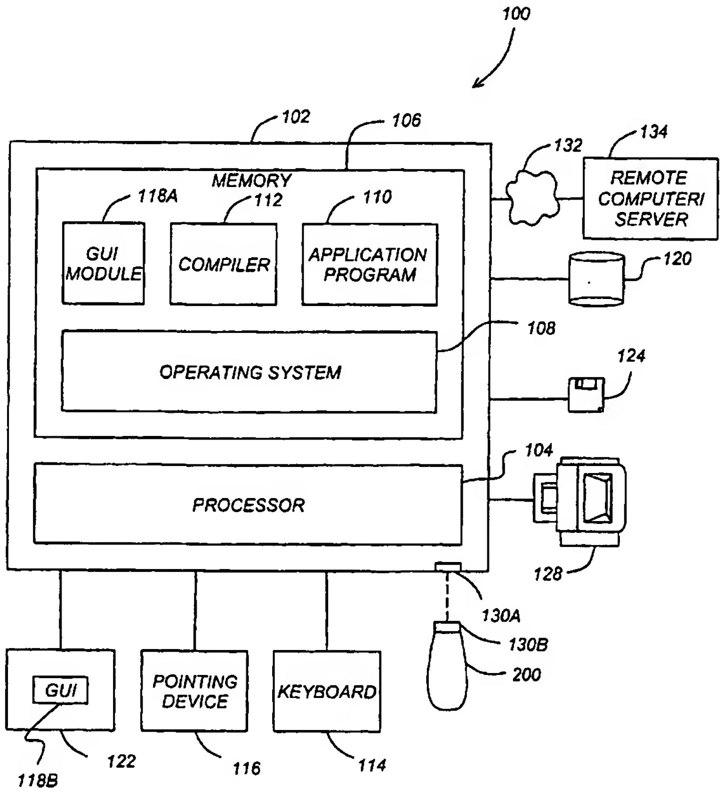
(74) Agents: **SMITH, Samuel, Leonard** et al.; J.A. Kemp & Co., 14 South Square, Gray's Inn, London WC1R 5JJ (GB).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European

[Continued on next page]

(54) Title: USB-COMPLIANT PERSONAL KEY USING A SMARTCARD PROCESSOR AND A SMARTCARD READER EMULATOR



(57) Abstract: A compact, self-contained, personal key is disclosed. The personal key comprises a USB-compliant interface releaseably coupleable to a host processing device operating under command of an operating system; a smartcard processor having a smartcard processor-compliant interface of communicating according to a smartcard input and output protocol; and an interface processor, communicatively coupled to the USB-compliant interface and to the smartcard processor-compliant interface, the interface processor implementing a translation module for interpreting USB-compliant messages into smartcard processor-compliant messages and for interpreting smartcard processor-compliant messages into USB-compliant messages.

patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

USB-COMPLIANT PERSONAL KEY USING A SMARTCARD PROCESSOR AND A SMARTCARD READER EMULATOR

— CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation-in-part of U.S. Patent Application No. 09/449,159, filed November 24, 1999, by Shawn D. Abbott, Bahram Afghani, Mehdi Sotoodeh, Norman L. Denton III, and Calvin W. Long, and entitled "USB-Compliant Personal Key with Integral Input and Output Devices," which is a continuation-in-part of U.S. Patent Application No. 09/281,017, filed March 30, 1999 by Shawn D. Abbott, Bahram Afghani, Allan D. Anderson, Patrick N. Godding, Maarten G. Punt, and Mehdi Sotoodeh, and entitled "USB-Compliant Personal Key," which claims benefit of U.S. Provisional Patent Application No. 60/116,006, filed January 15, 1999 by Shawn D. Abbott, Barham Afghani, Allan D. Anderson, Patrick N. Godding, Maarten G. Punt, and Mehdi Sotoodeh, and entitled "USB-Compliant Personal Key," all of which applications are hereby incorporated by reference herein.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to computer peripherals, and in particular to an inexpensive USB-compliant personal key that is compatible with existing smartcard processors, drivers, and instruction sets.

2. Description of the Related Art

In the last decade, the use of personal computers in both the home and in the office have become widespread. These computers provide a high level of functionality to many people at a moderate price, substantially surpassing the performance of the large mainframe computers of only a few decades ago. The trend is further evidenced by the increasing popularity of laptop and notebook computers, which provide high-performance computing power on a mobile basis.

The widespread availability of personal computers has had a profound impact on interpersonal communications as well. Only a decade ago, telephones or fax machines offered virtually the only media for rapid business communications. Today, a growing number of businesses and individuals communicate via electronic mail (e-mail). Personal computers have also been instrumental in the emergence of the Internet and its growing use as a medium of commerce.

While certainly beneficial, the growing use of computers in personal communications, commerce, and business has also given rise to a number of unique challenges. These challenges include the prevention of unauthorized use of software, ensuring the security of e-mail and other electronic communications, as well as Internet commerce.

Smartcards represent a longstanding attempt to deal with at least some of the foregoing challenges. Substantial resources have been made in the design and development of smartcards, smartcard readers, and the associated reader/smartcard drivers which allow computer applications to interface with the smartcard to perform security and data storage functions. Even so, smartcards have not enjoyed widespread popularity. Smartcard readers are relatively expensive, and not widely available. Further, the lack of uniform smartcard/smartcard reader physical interface standards have resulted in smartcard/smartcard reader physical interface compatibility problems, many of which remain unresolved.

USB-compliant personal keys, such as that which is disclosed in co-pending and commonly assigned U.S. Patent Application Nos. 09/449,159 and 09/281,017, described above, offer the benefit of smartcard functionality in a universally accepted USB form factor. The Universal Serial Bus (USB) is a connectivity standard developed by computer and telecommunication industry members for interfacing computers and peripherals. USB-compliant devices allow the user to install and hot-swap devices without long installation procedures and reboots, and features a 127 device bus capacity, dual-speed data transfer, and can provide limited power to devices attached on the bus. Because the USB connectivity standard is rapidly becoming available on most personal computers, it offers a standard, widely available physical interface, the unavailability of which has prevented smartcards from achieving widespread acceptance.

While smartcards have not enjoyed widespread popularity in the United States, they are widely accepted in Europe. Hence, many software applications and drivers have been developed for existing smartcard-based devices and their readers. Unfortunately, smartcard interface protocols such as those described in ISO 7816 are incompatible with the USB protocols used in the above-described devices. This incompatibility has led to two unfortunate consequences. First, to comply with USB interface protocol requirements, current USB-compliant personal keys utilize special purpose processors, instead of the low cost, limited capability processors currently available for smartcards. This increases the cost of the USB-compliant personal key, making widespread acceptance more difficult. Also, because each USB-compatible personal key may use a different processor (and different instruction sets), users may require different device drivers for different personal keys. This too represents another barrier to widespread acceptance of the personal key.

From the foregoing, it is apparent that there is a need for a USB-compliant personal key that is usable with legacy personal identification devices, such as processors having smartcard processors and/or those complying with the ISO 7816. There is also a need for a USB-compliant personal key that makes maximum use of existing smartcard protocols, software and devices wherever possible, and which retain at least a limited compatibility with existing devices designed to interface with smartcards. The present invention satisfies that need.

SUMMARY OF THE INVENTION

The present invention satisfies all of these needs with a personal key in a form factor that is compliant with a commonly available I/O interface such as the Universal Serial Bus (USB) and at the same time, usable with existing smartcard software applications. The personal key comprises a USB-compliant interface releaseably coupleable to a host processing device operating under command of an operating system; a smartcard processor having a smartcard processor-compliant interface for communicating according to a smartcard input and output protocol; and an interface processor, communicatively coupled to the USB-compliant interface and to the smartcard processor-compliant interface, the interface processor implementing a translation module for interpreting USB-compliant messages into smartcard processor-compliant messages and for interpreting smartcard processor-compliant messages into USB-compliant messages.

In one embodiment, the method comprises the steps of accepting a message comprising a smartcard reader command selected from a smartcard reader command set from a host computer operating system in a virtual smartcard reader; packaging the message for transmission via a USB-compliant interface according to a first message transfer protocol; transmitting the packaged message to a personal key communicatively coupled to the USB-compliant interface; receiving the packaged message in the personal key; unpackaging the message in the personal key to recover the smartcard reader command; translating the smartcard reader command into a smartcard command within the personal key; and providing the smartcard command to the smartcard processor.

The present invention is well suited for controlling access to network services, or anywhere a password, cookie, digital certificate, or smartcard might otherwise be used, including:

- Remote access servers, including Internet protocol security (IPSec), point to point tunneling protocol (PPTP), password authentication protocol (PAP), challenge handshake authentication protocol (CHAP), remote access dial-in user service (RADIUS), terminal access controller access control system (TACACS);
- Providing Extranet and subscription-based web access control, including hypertext transport protocol (HTTP), secure sockets layer (SSL);

- Supporting secure online banking, benefits administration, account management;
- Supporting secure workflow and supply chain integration (form signing);
- Preventing laptop computer theft (requiring personal key for laptop operation);
- Workstation logon authorization;
- Preventing the modification or copying of software;
- Encrypting files;
- Supporting secure e-mail, for example, with secure multipurpose Internet mail extensions (S/MIME), and open pretty good privacy (OpenPGP)
- Administering network equipment administration; and
- Electronic wallets, with, for example, secure electronic transaction (SET, MilliCent, eWallet)

BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

FIG. 1 is a diagram showing an exemplary hardware environment for practicing the present invention;

FIG. 2 is a block diagram of a personal key communicatively coupled to a host computer;

FIG. 3 is a block diagram of a personal key with a smartcard processor communicatively coupled to a host computer; and

FIGs. 4A-4D are flow charts presenting exemplary method steps that can be used to practice the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

In the following description, reference is made to the accompanying drawings which form a part hereof, and which is shown, by way of illustration, several embodiments of the present invention. It is understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

FIG. 1 illustrates an exemplary computer system 100 that could be used to implement the present invention. The host computer 102 comprises a processor 104 and a memory, such as random access memory (RAM) 106. The host computer 102 is operatively coupled to a display 122, which presents images such as windows to the user on a graphical user interface 118B. The host computer 102 may be coupled to other devices, such as a keyboard 114, a mouse device 116, a printer 128, etc. Of course, those skilled in the art will recognize that any combination of the above components, or any number of different components, peripherals, and other devices, may be used with the host computer 102.

Generally, the host computer 102 operates under control of an operating system 108 stored in the memory 106, and interfaces with the user to accept inputs and commands and to present results through a graphical user interface (GUI) module 118A. Although the GUI module 118A is depicted as a separate module, the instructions performing the GUI functions can be resident or distributed in the operating system 108, the computer program 110, or implemented with special purpose memory and processors. The host computer 102 also implements a compiler 112 which allows an application program 110 written in a programming language such as COBOL, C++, FORTRAN, or other language to be translated into processor 104 readable code. After completion, the application 110 accesses and manipulates data stored in the memory 106 of the host computer 102 using the relationships and logic that are generated using the compiler 112. The host computer 102 also comprises an input/output (I/O) port for a personal token 200 (hereinafter alternatively referred to also as a personal key 200). In one embodiment, the I/O port is a USB-compliant interface comprising a host computer USB-compliant interface 130A and a personal token USB-compliant interface 130B (hereinafter referred to collectively as the USB-compliant interface 130.

In one embodiment, instructions implementing the operating system 108, the computer program 110, and the compiler 112 are tangibly embodied in a computer-readable medium, e.g., data storage device 120, which could include one or more fixed or removable data storage devices, such as a zip drive, floppy disc drive 124, hard drive, CD-ROM drive, tape drive, etc. Further, the operating system 108 and the computer program 110 are comprised of instructions which, when read and executed by the computer 102, causes the computer 102 to perform the steps necessary to implement and/or use the present invention. Computer program 110 and/or operating instructions may also be tangibly embodied in memory 106 and/or data communications devices, thereby making a computer program product or article of manufacture according to the invention. As such, the terms "article of manufacture" and "computer program product" as used herein are intended to encompass a computer program accessible from any computer readable device or media.

The host computer 102 may be communicatively coupled to a remote computer or server 134 via communication medium 132 such as a dial-up network, a wide area network (WAN), local area network (LAN), virtual private network (VPN) or the Internet. Program instructions for computer operation, including additional or alternative application programs can be loaded from the remote computer/server 134. In one embodiment, the computer 102 implements an Internet browser, allowing the user to access the world wide web (WWW) and other internet resources.

Those skilled in the art will recognize that many modifications may be made to this configuration without departing from the scope of the present invention. For example, those skilled in the art will recognize that any combination of the above components, or any number of different components, peripherals, and other devices, may be used with the present invention.

FIG. 2 is a block diagram illustrating the components of one embodiment of a personal key 200. The personal key 200 communicates with and obtains power from the host computer 102 through a USB-compliant communication path in the USB-compliant interface 130 which includes the input/output port 130A of the host computer 102 and a matching input/output (I/O) port 130B on the personal key 200. The processor 212 is communicatively coupled to a memory 214, which stores data and instructions to implement the above-described features of the invention. In one embodiment, the memory 214 is a non-volatile random-access memory that can retain factory-supplied data as well as customer-supplied application related data. The processor 212 may also include some internal memory for performing some of these functions.

The processor 212 is optionally communicatively coupled to an input device 218 via an input device communication path 224 and to an output device 222 via an output device communication path 224, both of which are distinct from the USB-compliant interface 130. These separate communication paths 220 and 224 allow the user to view information about processor 212 operations and provide input related to processor 212 operations without allowing a process or other entity with visibility to the USB-compliant interface 130 to eavesdrop or intercede. This permits secure communications between the key processor 212 and the user. In one embodiment of the invention set forth more fully below, the user communicates directly with the processor 212 by physical manipulation of mechanical switches or devices actuatable from the external side of the key (for example, by pressure-sensitive devices such as buttons and mechanical switches). In another embodiment of the invention set forth more fully below, the input device includes a wheel with tactile detents indicating the selection of characters.

The input device and output devices 218, 222 may cooperatively interact with one another to enhance the functionality of the personal key 200. For example, the output device 222 may provide information prompting the user to enter information into the input device 218. For example, the output device 222 may comprise a visual display such as an alphanumeric LED or LCD display (which can display Arabic numbers and or letters) and/or an aural device. The user may be prompted to enter information by a beeping of the aural device, by a flashing pattern of the LED, or by both. The output device 222 may also optionally be used to confirm entry of information by the input device 218. For example, an aural output device may beep when the user enters information into the input device 218 or when the user input is invalid. The input device 218 may take one of many forms, including different combinations of input devices.

Although the input device communication path 220 and the output device communication path 224 are illustrated in FIG. 2 as separate paths, the present invention can be implemented by combining the paths 220 and 224 while still retaining a communication path distinct from the USB-compliant interface 130. For example, the input device 218 and output device 222 may be packaged in a single device and communications with the processor 212 multiplexed over a single communication path.

FIG. 3 is a block diagram of the personal key 200 and host computer 102 as applied to the present invention. Unlike the personal key 200 illustrated in FIG. 2, the personal key 300 illustrated in FIG. 3 comprises a smartcard processor 320. The smartcard processor 300 is a processor which complies with well-known smartcard I/O protocols and smartcard command sets and functions, such as those described by the International Standards Organization (ISO) standard 7816 Part III (defining electronic properties and transmission characteristics), which is hereby incorporated by reference herein.

Physically, the smartcard compliant I/O interface 324 includes a serial I/O line, a reset (RST) line, a clock (CLK) line, a programming voltage (VPP), a power supply voltage (VCC) and a ground. This I/O interface 324 is further described in the publication "Introduction to Smartcards" by Dr. David B. Everett, which was published in 1999 by the Smart Card News Ltd., and is incorporated by reference herein.

As was the case with the personal key 200 and host computer 102 illustrated in FIG. 1, the present invention allows the use of a personal key 300 communicating with the host computer 102 via a USB-compliant interface 130. However, the substitution of the smartcard processor 320 for the ordinary processor 212 depicted in

FIG. 2 has several advantages. First, smartcard processors 212 are relatively inexpensive and readily available. Second, a large number of application programs 110 have been developed for the use of smartcards, including the personal computer/smartcard (PC/SC) interface developed by the MICROSOFT CORPORATION. By providing a smartcard processor (which complies with the smartcard I/O protocols and supports smartcard command sets), this software can be used with a personal key 300 in a USB-compliant form factor. .
The use of the smartcard processor 320 in the personal key 300 is enabled by use of an interface processor 314 communicatively coupled to the smartcard processor 320 via a smartcard-compatible (S/C 7816) interface 324. The interface processor 314 comprises a smartcard reader emulator module (SREM) 316 and a translation module 318. The SREM 316 implements functions that emulate those of a smartcard reader, thus projecting the image of a smartcard reader to the smartcard processor 320. The SREM 316 provides all instructions and commands to the smartcard processor 320 and receives messages and responses from the smartcard processor 320 according to the S/C protocol.

The host computer 102 comprises a virtual smartcard reader module (VSRM) 302. The VSRM comprises a communication module 312, an answer-to-reset module 308, and a smartcard insertion/removal reporting module 306. The communication module 312 packages messages intended for the personal key 300 for transmission via the USB-compliant interface. In one embodiment, messages and commands that are sent to the personal key 300 packaged as:

USB command = USB header + USB cdata (wherein USB cdata is the smartcard compliant command)

and messages and responses from the personal key 300 are packaged as:

USB response = USB header + USB rdata (wherein USB rdata is the smartcard compliant response)

These packaged messages are unpacked by the translation module 318 in the personal key 300. Similarly, messages transmitted by the smartcard processor 320 to the host computer 102 are packaged by the translation module 318 and unpackaged by the communication module 312 before being provided to the operating system 108, the application program interface 260, and the application 110 using the personal key 300 to perform operations.

Just as the SREM 316 emulates the presence of a smartcard reader for the smartcard processor 320, the VSRM 302 emulates the presence of a smartcard reader to the OS 108 in the host computer 102. These functions are accomplished in the bootup module 311, the insert/remove module 306, the answer-to-reset module 308, and the PTS module 310.

As a part of a normal bootup sequence, the host computer's 102 operating system performs a startup sequence to determine which hardware elements are available for use. In prior art smartcard systems, the smartcard reader remains coupled to the host computer 102, whether a smartcard is inserted into the reader or not. Hence, the smartcard reader can respond to startup sequence queries, and the smartcard reader is recognized by the operating system 108 for further operations. However, in the present invention, there is no smartcard reader to answer to the bootup query, and the operating system would ordinarily be unable to operate with a smartcard thereafter. To solve this problem, the present invention comprises a bootup module 311, which responds to messages from the operating system 108 in the same way as a smartcard reader would if it were coupled to the host computer 102.

Similarly, the insert/remove module 306 provides an indication to the operating system 108 that the personal key 300 has been inserted or removed from the

USB-compliant interface 130. This is accomplished by querying the host computer USB-compliant interface port 130A.

When a software application calls 110, via API 260 and the operating system 108 invokes a command that calls for a smartcard related function, the smartcard reader passes a reset command to the smartcard. The smartcard returns an answer-to-reset message which indicates, among other things, the protocol and I/O interface supported by the attached smartcard.

The reset signal is used to start up the program contained in a memory 322 communicatively coupled to or resident within the smartcard processor 320. The ISO standard defines three reset modes, internal reset, active low reset, and synchronous high active reset. Most smartcard processors 320 operate using the active low reset mode. In this mode, the smartcard processor 320 transfers control to the entry address for the program when the reset signal returns to the high voltage level. The synchronous mode of operation is more commonly met with smartcards used for telephonic applications.

The sequence of operations for activating the smartcard processor 320 is defined in order to minimize the possibility of damaging the smartcard processor 320. Of particular importance is avoiding corruption of the non-volatile memory 322 of the smartcard. Most smartcard processors 320 operate using an active low reset mode in which the smartcard processor 320 transfers control to the entry address for the program when the reset signal returns to the high voltage level. The sequence performed by the smartcard processor includes the steps of setting the RST line low, applying VCC to the proper supply voltage, setting the I/O in the receive mode, setting VPP in the idle mode, applying the clock, and taking the RST line high (active low reset).

In prior art smartcard systems, after the reset signal is applied by the smartcard reader, the smartcard processor 320 responds with an answer-to-reset message. For the active low reset mode, the smartcard processor 320 should respond between 400 and 40,000 clock cycles after the rising edge of the reset signal. The answer-to-reset signal is at most 33 characters, and includes 5 fields including an initial character (TS), a format character (TO), interface characters (TAi, TBi, TCi, and TDi), historical characters (T1, T2, ... , TK), and a check character (TCK). Among other things, the answer-to-reset signal provides an indication of the smartcard protocol(s) which are supported smartcard processor. Typical smartcard protocols include the T=0 protocol (asynchronous half duplex byte transmission) and T=1 (asynchronous half duplex block transmission).

In the embodiment of the present invention shown in FIG. 3, the reset signal is provided by the VSRM 302, packaged by the communication module 312, and sent via the USB-compliant interface 130B to the personal key 300. The message is unwrapped by the translation module 318. Then, the smartcard reader emulation module activates the RST signal path in the smartcard interface 324, thus providing the RST command to the smartcard processor 320. The smartcard processor 320 responds with an answer-to-reset message, sends the message via the serial I/O line of the smartcard interface 324 to the interface processor 314. The message is then packaged by the translation module 318 and transmitted to the host computer 102 via the USB-compliant interface 326. The message is then unpackaged by the communication module 312 and provided to the operating system 108 and ultimately, the application 110 that requested the use of the smartcard.

In another embodiment of the present invention, the personal key 300 does not comprise a smartcard processor 320, but rather a special purpose processor which does not respond to messages and commands in the smartcard I/O protocol (such as that which is illustrated in FIG. 1). The present invention can still be used with existing smartcard applications 110, however, because the VSRM 302 and the interface processor 314 can be used to simulate the presence of a smartcard processor 320. When the smartcard software application 110 desires use of the personal key 300, the VSRM accepts the reset command from the PC/SC modules in the operating system 108, translates the reset message into a functionally equivalent message for the special purpose processor in the personal key 300, and transmits the message to the personal key 300. After the personal key 300 is activated, it sends a message indicating as such to the host computer 102. The VSRM 302, and translates this message to a response that is compatible with the smartcard application 110, namely, an ATR message. Alternatively, the smartcard command to special purpose processor command translation can occur in the emulation processor 314 in the personal key 300.

Returning to the embodiment disclosed in FIG. 3, after the smartcard processor has issued the ATR message, a protocol type selection (PTS) message may be sent to the smartcard processor 320. The PTS message from the OS 108 is received by the PTS module 310 in the VSRM 302, packaged for transmission via the USB-compliant interface 130 to the personal key 300, where it is unpackaged and provided to the smartcard processor 320. The smartcard provides a response consistent with the ISO standards to the emulation module 316. The response is packaged, and transmitted over the USB-compliant interface 130 to the host computer 102, where it is unpackaged by the communication module 312 and provided to the operating system.

FIGs. 4A-4D are flow charts presenting exemplary method steps used to practice one embodiment of the present invention. When the host computer 102 is booted up, the virtual smartcard reader 302 accepts 402 a bootup query from the host computer's operating system 108. Although a smartcard reader is not communicatively coupled to the host computer 130 the virtual smartcard reader 302 emulates the existence of a smartcard reader and provides an indication that a smartcard reader is available to the OS 108. Consequently, when the bootup procedures are completed, a smartcard reader will be registered as an available device to smartcard applications 110.

When the host computer is booted up, a personal key 300 may or may not be communicatively coupled to the USB-compliant interface 130. When a personal key 300 is not attached, the VSRM 302 provides 404 the same indication to the operating system 108 as would be supplied by a smartcard reader without an inserted smartcard. This is accomplished by receiving 406 an indication that the personal key has been communicatively coupled to the USB-compliant interface, and providing an indication to the host computer operating system. Since the VSRM is emulating the functions of a smartcard, the indication is provided 408 to the host computer operating system (or equivalently, the personal computer/smartcard (PC/SC) interface modules therein) is that of an insert event.

If desired and the smartcard processor 320 supports multiple protocols, a protocol type selection (PTS) command may be issued by the operating system 108. The VSRM 302 receives 410 the PTS command, packages the command for transmission to the personal key 300 via the USB-compliant interface 130. The wrapped PTS command is then transmitted over the USB-compliant interface 130 and received by the personal key 300. The PTS command is unwrapped by the translate module 318 in the interface processor 314 and provided to the smartcard processor 320 via the smartcard-compliant interface 324. The smartcard processor computes the appropriate response, sends the response to the interface processor 314, where the response is packaged by the translate module 318 for transmission to the host computer 102 via the USB-compliant interface 130. The communication module 312 unpackages the response, and the PTS module 310 formats the response, if necessary, to be consistent with a PTS response received from a smartcard reader. The formatted response is then provided 412 to the OS 108.

FIG. 4B is a flow chart describing exemplary method steps used to provide commands and/or data from the OS 108 to the smartcard processor 320 and from the smartcard processor 320 to the OS 108. A message, which may comprise a smartcard reader command belonging to a smartcard reader command set is accepted 414 from a host computer operating system 108 in the virtual smartcard reader module (VSRM) 302. The message is packaged 416 for transmission via the USB-compliant interface 130 according to a first message transfer protocol.

The packaged message is then transmitted 418 to the communicatively coupled personal key 300 via the USB-compliant interface 130. The packaged message is received 420 and unpackaged 422 in the personal key 300. If the smartcard reader command requires additional processing before being forwarded to the smartcard processor 320, the smartcard reader command is translated 424 into a smartcard command within the personal key 300 before being provided 426 to the smartcard processor 320.

The smartcard processor 320 then performs the indicated operation, and a response is accepted 428 from the smartcard processor 320. If the smartcard response requires further processing by a smartcard reader, the smartcard response is translated 430 into a smartcard reader response. The smartcard reader response is then packaged 432 and transmitted 434 to the host computer 102 via the USB-compliant interface 130. The host computer 102 receives 436 and unpackages 438 the message and provides 440 the response to the smartcard software application 110 that issued the command.

Next, when the personal key 300 is removed, the VSRM 302 reports 444 an indication to the OS 108 that the "virtual smartcard" (the personal key 300) has been removed. The provided indication is the same as that which would be provided by a smartcard reader when a smartcard is removed. The indication can be obtained, for example by receiving 442 an indication from a USB driver or other device indicating the removal of a USB device.

In summary, Tables I and II provides an summary of the communication protocol for an OS 108 command from the host computer 102 to the smartcard processor 320 in the personal key (Table I), and for a smartcard processor 320 response to the operating system 108.

Step	Description
1	Smartcard reader command issued from OS 108 is passed to VSRM 302
2	VSRM 302 adds a USB header, and creates a USB command
3	VSRM's 302 communication module 312 sends the USB command to the personal key 300
4	The translation module 318 strips off the USB header and recovers the smartcard command
5	The smartcard command is sent to the smartcard processor 320
6	The smartcard processor 320 executes the function requested by the smartcard command

Table I

Step	Description
1	Smartcard processor 320 generates a smartcard response
2	The smartcard response is sent from the smartcard processor 320 to the translation module 318
3	The translation module 318 adds a USB header to create a USB response
4	The USB response is transmitted to the VSRM 302
5	The communication module 312 strips off the USB header and recovers the smartcard response
6	The smartcard response is transmitted to the OS 108

Table II

Tables III and IV provides a summary of the communication protocol for a request from an application program 110 to the smartcard processor 320 and for a request from an application program 110 to the smartcard processor 320.

Step	*Description*
1	Smartcard processor 320 command from the application program 110 is sent to the OS 108 via an API 260
2	The smartcard processor 320 command is sent from the OS 108 to the VSRM 302
3	The VSRM 302 adds a USB header to the smartcard processor 320 command to create a USB-compatible command
4	The VSRM's comm module 312 sends the USB-compliant command to the personal key 300
5	Translation module 318 strips off the USB header and recovers the smartcard processor command
6	The smartcard processor command is transmitted to the smartcard processor 320
7	The smartcard processor 320 performs the function indicated by the smartcard processor command

Table III

Step	Description
1	The smartcard processor 320 generates a response to the smartcard processor command
2	The response is provided to the translation module 318
3	The translation module adds a USB header to create a USB-compatible smartcard processor response
4	The USB-compatible smartcard processor response is sent to the VSRM 302
5	The communication module 312 strips off the USB header to recover the smartcard processor response
6	The smartcard processor response is provided to the application 110 via the OS 108 and the API 260

Table IV

Conclusion

This concludes the description of the preferred embodiments of the present invention. In summary, the present invention describes a personal key comprising a USB-compliant interface releaseably coupleable to a host processing device operating under command of an operating system; a smartcard processor having a smartcard processor-compliant interface for communicating according to a smartcard input and output protocol; and an interface processor, communicatively coupled to the USB-compliant interface and to the smartcard processor-compliant interface, the interface processor implementing a translation module for interpreting USB-compliant messages into smartcard processor-compliant messages and for interpreting smartcard processor-compliant messages into USB-compliant messages. In another embodiment, the invention is described by a method comprising the steps of accepting a message comprising a smartcard reader command selected from a smartcard reader command set from a host computer operating system in a virtual smartcard reader; packaging the message for transmission via a USB-compliant interface according to a first message transfer protocol; transmitting the packaged message to a personal key communicatively coupled to the USB-compliant interface; receiving the packaged message in the personal key; unpackaging the message in the personal key to recover the smartcard reader command; translating the smartcard reader command into a smartcard command within the personal key; and providing the smartcard command to the smartcard processor.

The foregoing description of the preferred embodiment of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the invention be limited not by this detailed description, but rather by the claims appended hereto. The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.

WHAT IS CLAIMED IS:

1. A compact personal token (300), comprising:

a USB-compliant interface (130B) releaseably coupleable to a host processing device (102) operating under command of an operating system (108);

a smartcard processor (320) having a smartcard processor-compliant interface (324) for communicating according to a smartcard input and output protocol;

an input device (218) communicatively coupled to the smartcard processor for providing secure input to the processor;

an interface processor (314), communicatively coupled to the USB-compliant interface (130B) and to smartcard processor-compliant interface (324) the interface processor (314) implementing a translation module (318) for interpreting USB-compliant messages into smartcard processor-compliant messages and for interpreting smartcard processor-compliant messages into USB-compliant messages.

2. The apparatus of claim 1, wherein the interface processor (314) emulates a smartcard reader to the smartcard processor (320).

3. The apparatus of claim 1, wherein:

the host processing device (102) comprises a virtual smartcard reader in communication with the operating system, the virtual smartcard reader for emulating a smartcard reader communicatively coupled to the host processing device (102) and including a communication module (312) for packaging messages for transmission to the personal token (300) via the USB compliant interface (130) according to a first protocol and for unpackaging messages received from the personal token (300) via the USB-compliant interface according to the first protocol; and

the interface processor translation module (318) unpackages messages from the host processing device (102) according to the first protocol and packages messages destined for the host processing device (102) according to the first protocol.

4. The apparatus of claim 3, wherein the virtual smartcard reader further comprises a bootup module (311) for responding to an operating system bootup procedure with an indication that a smartcard reader is communicatively coupled to the host processor.

5. The apparatus of claim 3, wherein the virtual smartcard reader further comprises an answer-to-reset (ATR) module (308) for providing an ATR message to the operating system (108) in response to a reset message.

6. The apparatus of claim 3, wherein the virtual smartcard reader further comprises a reporting module for receiving and reporting the insertion of the personal token in a USB-compliant port communicatively coupled to the host processor (102) and the removal of the personal token as a removal of a smartcard from a smartcard reader.

7. The apparatus of claim 3, wherein the virtual smartcard reader further comprises a protocol selection module for receiving a protocol type selection (PTS) command from the operating system and providing a PTS response message to the operating system (108).

8. A method of communicating between a smartcard processor (320) in a personal key (300) communicatively coupled to a host computer (102) via a USB-compliant interface (130), comprising the steps of:

accepting a message comprising a smartcard reader command selected from a smartcard reader command set from a host computer operating system (108) in a virtual smartcard reader;

packaging the message for transmission via a USB-compliant interface (130) according to a first message transfer protocol;

transmitting the packaged message to a personal key (300) communicatively coupled to the USB-compliant interface (130);

receiving the packaged message in the personal key (300);

unpackaging the message in the personal key (300) to recover the smartcard reader command;

translating the smartcard reader command into a smartcard command within the personal key (300); and

providing the smartcard command to the smartcard processor (320);

accepting a user input to the smartcard processor (320) via an input device (218) communicatively coupled to the smartcard processor (320) via an input communication device communication path distinct from the USB-compliant interface (130);

accepting a smartcard response from the smartcard processor (320);

translating the smartcard response into a smartcard reader response;

packaging the smartcard reader response for transmission to the host processor (102) via the USB-compliant interface (130);

transmitting the packaged message from the personal key (300) to the host processor (102);

receiving the packaged message in the host computer (102);

unpackaging the smartcard reader response; and

providing the smartcard reader response to the host processor operating system (108).

9. The method of claim 8, further comprising the steps of:

accepting a startup query from the host computer operating system (108) in the virtual smartcard reader; and

providing an indication that a smartcard reader is communicatively coupled to the host computer to the host computer operating system (108).

10. The method of claim 9, further comprising the steps of:

receiving an indication that the personal key (300) has been communicatively coupled to the USB-compliant interface (130);

reporting the indication that the personal key (300) is communicatively coupled to the USB-compliant interface (130) to the host processor operating system (108) as the insertion of a smartcard;

receiving an indication that the personal key (300) has been communicatively decoupled from the USB-compliant interface (130); and

reporting the indication that the personal key has been communicatively decoupled from the USB-compliant interface (130) to the host processor operating system (108) as the removal of the smartcard.

11. The method of claim 8, further comprising the steps of:

receiving a protocol type selection (PTS) command from the host computer operating system (108); and

providing a PTS response message to the operating system (108).

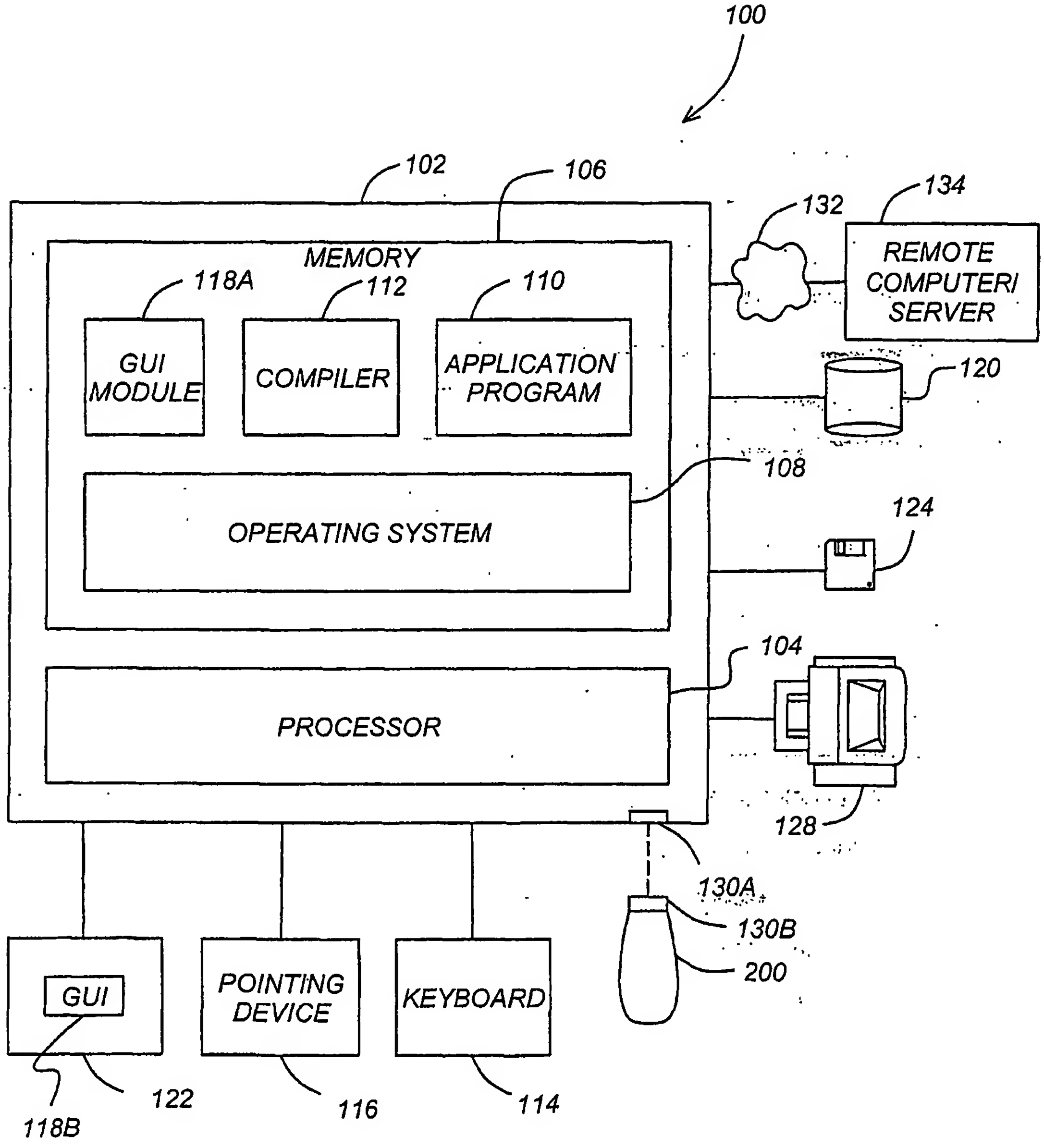

100
102
106
MEMORY
118A
112
110
GUI MODULE
COMPILER
APPLICATION PROGRAM
108
OPERATING SYSTEM
104
PROCESSOR
132
134
REMOTE COMPUTERI SERVER
120
124
128
130A
130B
200
GUI
POINTING DEVICE
KEYBOARD
118B
122
116
114

FIG. 1

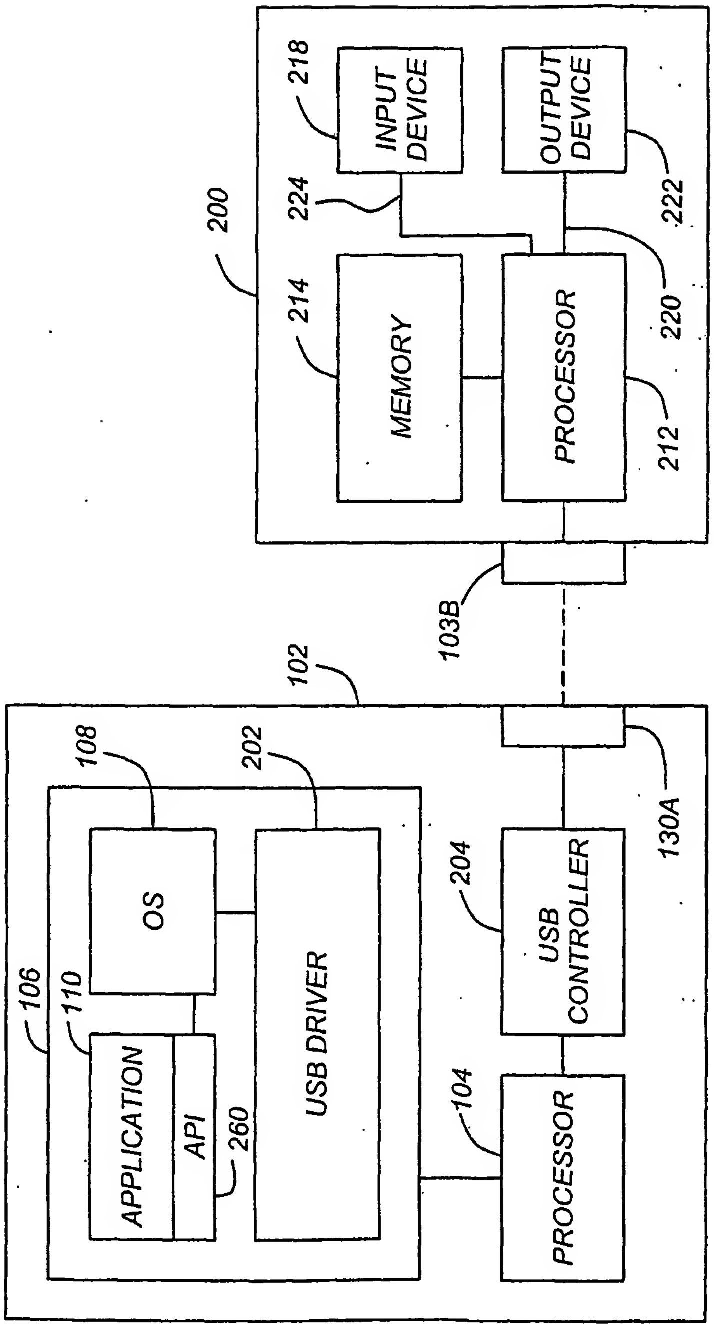
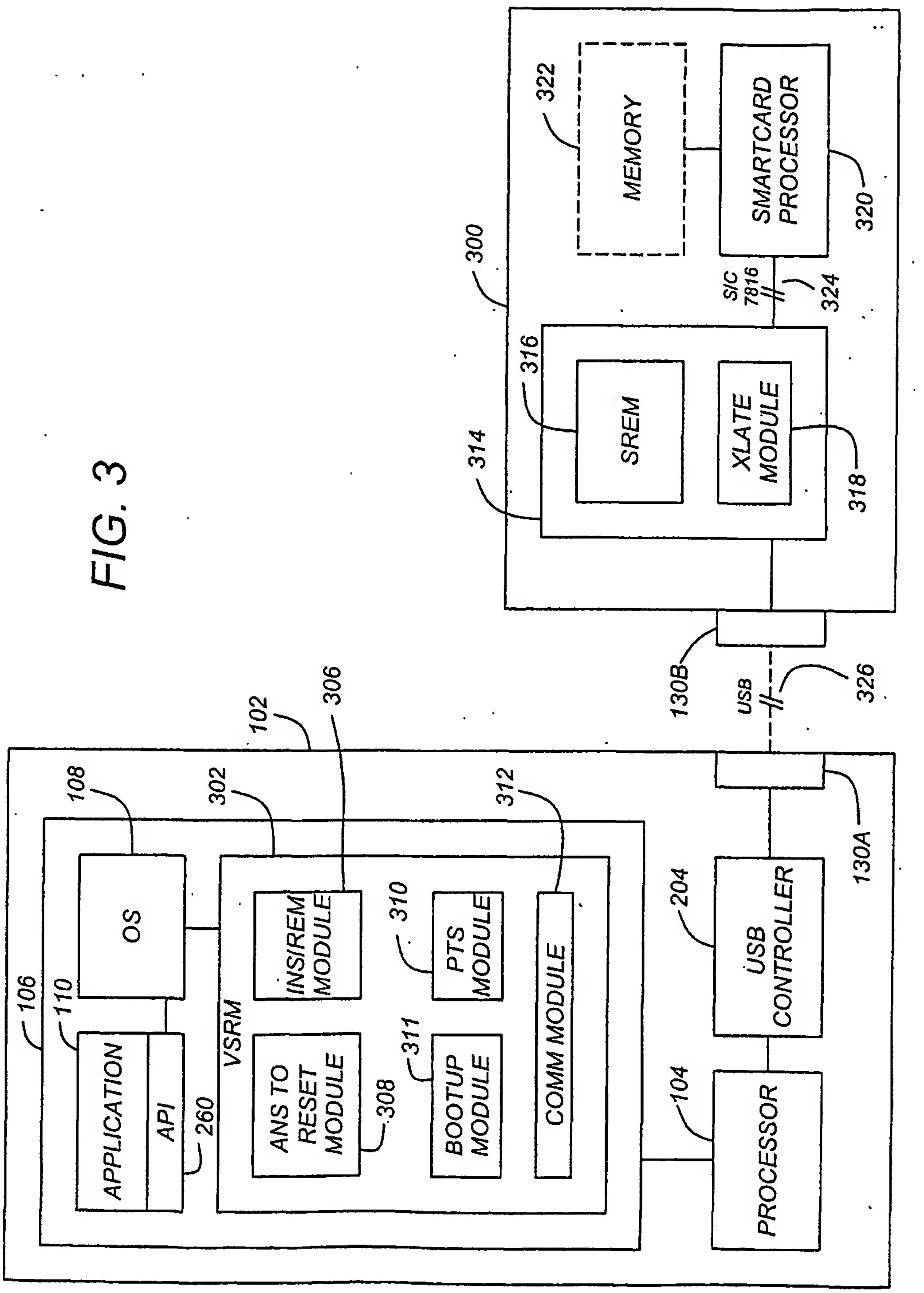

200
218
INPUT DEVICE
OUTPUT DEVICE
224
222
214
MEMORY
PROCESSOR
220
212
103B
102
108
202
OS
USB DRIVER
106
110
APPLICATION
API
260
204
USB CONTROLLER
130A
104
PROCESSOR

FIG. 2

FIG. 3


300
322
MEMORY
SMARTCARD PROCESSOR
320
S/C 7816
324
316
SREM
314
XLATE MODULE
318
130B
USB
326
102
306
108
302
312
OS
INS/REM MODULE
310
PTS MODULE
COMM MODULE
106
110
APPLICATION
API
260
VSRM
ANS TO RESET MODULE
308
311
BOOTUP MODULE
204
USB CONTROLLER
130A
104
PROCESSOR

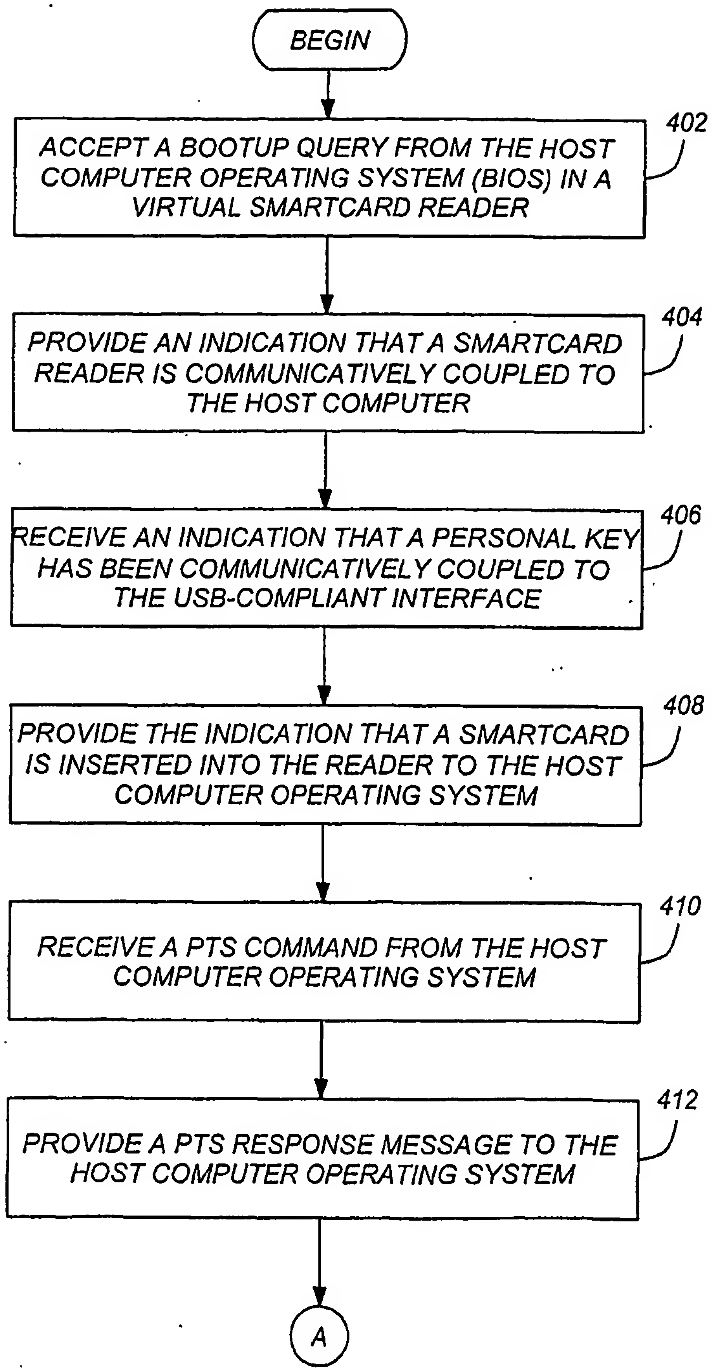

BEGIN
ACCEPT A BOOTUP QUERY FROM THE HOST COMPUTER OPERATING SYSTEM (BIOS) IN A VIRTUAL SMARTCARD READER
402
PROVIDE AN INDICATION THAT A SMARTCARD READER IS COMMUNICATIVELY COUPLED TO THE HOST COMPUTER
404
RECEIVE AN INDICATION THAT A PERSONAL KEY HAS BEEN COMMUNICATIVELY COUPLED TO THE USB-COMPLIANT INTERFACE
406
PROVIDE THE INDICATION THAT A SMARTCARD IS INSERTED INTO THE READER TO THE HOST COMPUTER OPERATING SYSTEM
408
RECEIVE A PTS COMMAND FROM THE HOST COMPUTER OPERATING SYSTEM
410
PROVIDE A PTS RESPONSE MESSAGE TO THE HOST COMPUTER OPERATING SYSTEM
412
A

FIG. 4A

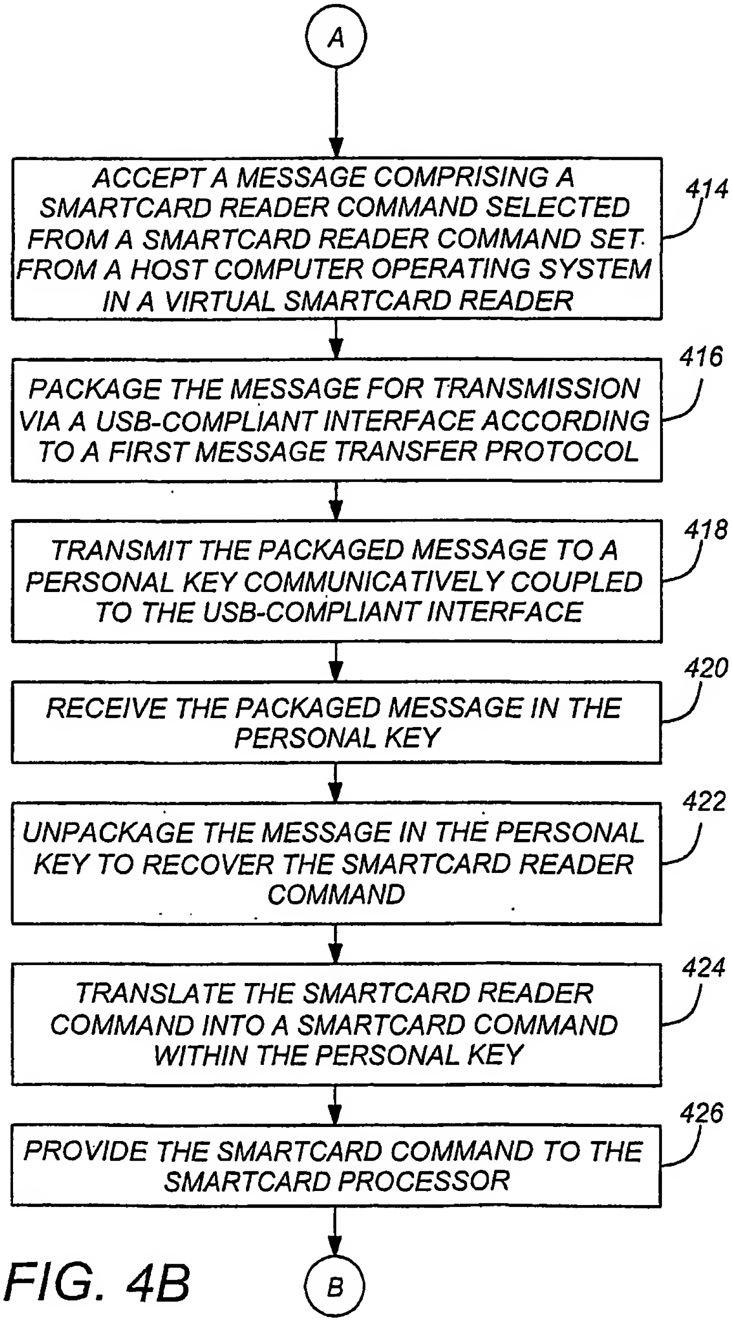

A
ACCEPT A MESSAGE COMPRISING A SMARTCARD READER COMMAND SELECTED FROM A SMARTCARD READER COMMAND SET FROM A HOST COMPUTER OPERATING SYSTEM IN A VIRTUAL SMARTCARD READER
414
PACKAGE THE MESSAGE FOR TRANSMISSION VIA A USB-COMPLIANT INTERFACE ACCORDING TO A FIRST MESSAGE TRANSFER PROTOCOL
416
TRANSMIT THE PACKAGED MESSAGE TO A PERSONAL KEY COMMUNICATIVELY COUPLED TO THE USB-COMPLIANT INTERFACE
418
RECEIVE THE PACKAGED MESSAGE IN THE PERSONAL KEY
420
UNPACKAGE THE MESSAGE IN THE PERSONAL KEY TO RECOVER THE SMARTCARD READER COMMAND
422
TRANSLATE THE SMARTCARD READER COMMAND INTO A SMARTCARD COMMAND WITHIN THE PERSONAL KEY
424
PROVIDE THE SMARTCARD COMMAND TO THE SMARTCARD PROCESSOR
426
B

FIG. 4B

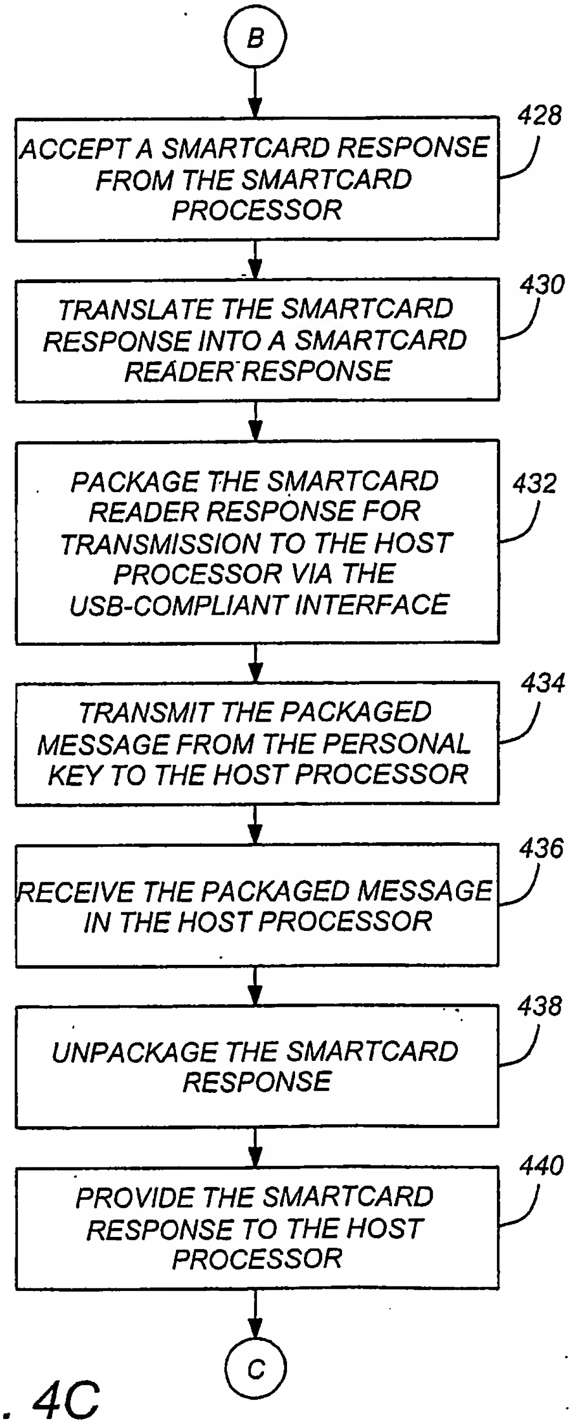

B
ACCEPT A SMARTCARD RESPONSE FROM THE SMARTCARD PROCESSOR
428
TRANSLATE THE SMARTCARD RESPONSE INTO A SMARTCARD READER RESPONSE
430
PACKAGE THE SMARTCARD READER RESPONSE FOR TRANSMISSION TO THE HOST PROCESSOR VIA THE USB-COMPLIANT INTERFACE
432
TRANSMIT THE PACKAGED MESSAGE FROM THE PERSONAL KEY TO THE HOST PROCESSOR
434
RECEIVE THE PACKAGED MESSAGE IN THE HOST PROCESSOR
436
UNPACKAGE THE SMARTCARD RESPONSE
438
PROVIDE THE SMARTCARD RESPONSE TO THE HOST PROCESSOR
440
C

FIG. 4C

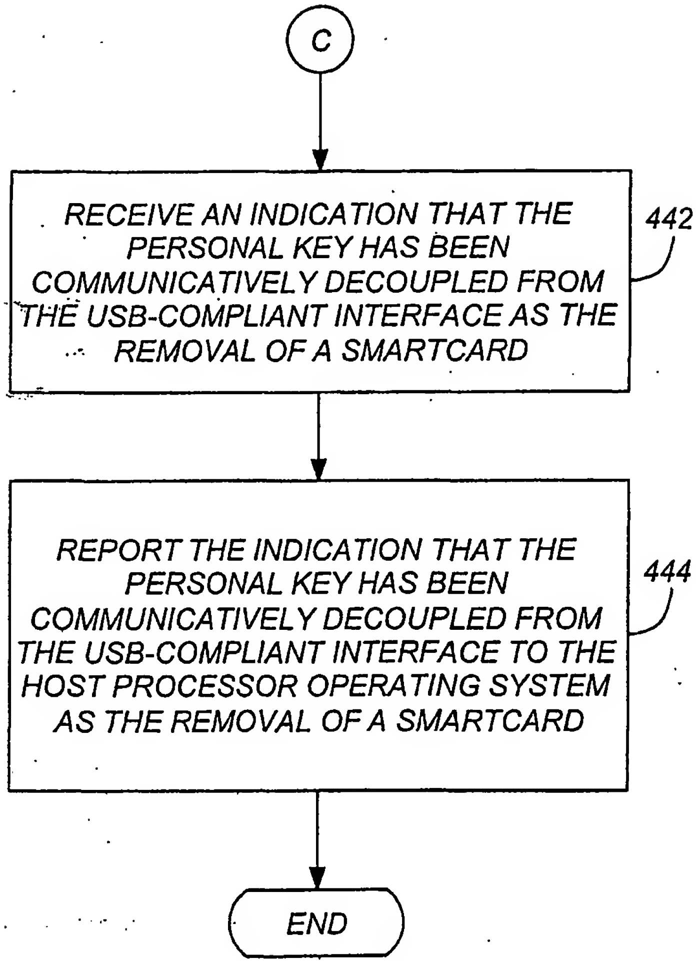

C
RECEIVE AN INDICATION THAT THE PERSONAL KEY HAS BEEN COMMUNICATIVELY DECOUPLED FROM THE USB-COMPLIANT INTERFACE AS THE REMOVAL OF A SMARTCARD
442
REPORT THE INDICATION THAT THE PERSONAL KEY HAS BEEN COMMUNICATIVELY DECOUPLED FROM THE USB-COMPLIANT INTERFACE TO THE HOST PROCESSOR OPERATING SYSTEM AS THE REMOVAL OF A SMARTCARD
444
END

FIG. 4D